REMARKS

The Examiner is thanked for the performance of a thorough search.

STATUS OF CLAIMS

Claim 21 has been cancelled.

Claims 1-9 and 11-19 have been amended.

Claims 22-31 have been added.

No claims have been withdrawn.

Claims 1-31 are currently pending in the application.

SUMMARY OF THE REJECTIONS/OBJECTIONS

Claims 1-9, 11-19, and 21 have been rejected under 35 U.S.C. § 103(a) as allegedly unpatentable over U.S. Patent Number 5,835,766 issued to Iba et al. (" *Iba* ") in view of U.S. Patent Number 5,778,179 issued to Kanai et al. (" *Kanai* "). Claims 1 and 11 have also been rejected under 35 U.S.C. § 103(a) as allegedly unpatentable over U.S. Patent Number 5,459,871 issued to Van Den Berg (" *Van Den Berg* ") in view of U.S. Patent Number 4,412,285 issued to Neches et al. (" *Neches* "). The rejections are respectfully traversed.

A. DISCUSSION OF CLAIM 1

As amended above, Claim 1 features:

"A method of determining participants of a **distributed transaction** in a distributed system, the method comprising the steps of:

registering, in a name service, participant data that identifies a plurality of participants that are participating in said distributed transaction, wherein **said step of registering occurs in response to said plurality of participants commencing participation in said distributed transaction**;

causing a node that requires information about participants in said distributed transaction to request said participant data from said name service." (emphasis added).

As amended, Claim 1 is in the context of a "**distributed transaction**," not a distributed operation. Also the feature in Claim 1 of "wherein **said step of registering occurs in response to said plurality of participants commencing participation in said distributed transaction**," is similar to previously presented, but now cancelled, Claim 21.

To aid in understanding Claim 1, consider the embodiment illustrated in Fig. 2 of the Application and described on pages 8 et seq., where a coordinator process for distributed transaction X 222 is responsible for coordinating other processes that are participating in distributed transaction X, such as slaves 232, 234, 236, and 252. Coordinator process 222 sends a publication request 224 to name service 202 to register the participants (e.g., slaves 232, 234, 236, and 252) for distributed transaction X. In response to publication request 224, name service 202 registers the participant data by storing replicated published data 219, 239, 259, 279 on database servers 210, 230, 250, 270, respectively.

As described in the Application, a "publication request is a request to make information available to a set of name service clients that request the information" and "the name service 202 supplies the data to any name service client requesting [the] data..." (Page 9, lines 5-12.) In order for the registered data to be available to the clients that request the data, the data can be registered at a number of times. For example, the data can be registered prior to the participants of distributed transaction X beginning their processing of distributed transaction X. As another example, the data can be registered as a result of coordinator process for distributed transaction X 222 commencing the processing of distributed transaction X, but after individual participants beginning processing distributed transaction X. In general, the participant data can be registered at any time prior to a client making a request for some or all of the registered participant data. As one example, the request can be made to ascertain the participants in distributed transaction X to determine whether a deadlock has occurred in a distributed database transaction.

B. CLAIM 1 & *IBA*

(1) DISCUSSION OF *IBA*'S APPROACH USING A GLOBAL DEADLOCK DETECTOR

Iba discloses a "system for detecting global deadlocks using wait-for graphs and identifiers of transactions related to the deadlocks in a distributed transaction processing system and a method of use therefore." (Title.) Specifically, *Iba* discloses an approach for using a global deadlock detector as part of a transaction manager to detect deadlocks based on wait-for relations that arise from global transaction that stretch over a plurality of resource managers. (Abstract; Fig. 6 and Fig. 7.) By including a deadlock detector as part of the transaction manager that manages the global transactions, *Iba* solves the problem of how to detect deadlocks stretching over resource managers between global transactions. (Col. 3, lines 25-30.) As described in the Background section of *Iba*, the conventional approach in this situation is to wait until a time-out occurs by monitoring execution time of one of the transactions whose processing has been substantially stopped due to a deadlock. (Col. 3, lines 31-35.)

Thus, the approach of *Iba* is to use the global deadlock detector with a wait-for graph (WFG) in the same manner as a local deadlock detector, namely to collect up wait-for requests from all of the global transactions, and then trace such requests to identify any loops that would indicate a deadlock. (See Col. 3, lines 43-50.) The major difference between the global deadlock detector and the local deadlock detector is that the former receives wait-for requests from all global transaction managed by the global transaction manager whereas the local deadlock detectors only receive wait-for requests for resources that are associated with the particular resource manager of which the local deadlock detector is a part. (See Col. 5, lines 11-24.)

Fig. 10A, Fig. 10B, Fig. 11 and the associated text of *Iba* illustrate the approach of using the global deadlock detector. "Fig. 10A shows the contents of node constituting a WFG and the contents comprise a chain portion and a SID portion as an identifier...Fig. 10B conceptually illustrates a tree structure of the chain portion." (Col. 10, lines 39-41 and 44-45.) Note that Fig. 10B lists the wait requests, such as "T2 is waiting for T1," etc. In Fig. 11, step S2 registers a request for registration of the wait-for relation of the global transaction.

(Col. 11, lines 37-44.) If the wait-for relation is newly registered, then step S3 of Fig. 11 sets that wait-for relation as the starting point to trace the loop for deadlock determination. (Col. 11, lines 44-48.) Steps S4, S5, and S6 describe determining whether a loop is detected, and if so, a transaction is selected to be cancelled in step S7, and if not, then the process returns to step S4. (Col. 11, lines 49-67.)

To summarize the approach of *Iba*, a global deadlock detector collects wait-for relations/requests from the global transactions. As each new wait request is received, a check is made to see if a loop can be detected. If so, a deadlock is identified and processing continues to clear the deadlock. If not, the process returns to wait for the next wait request. Note that in *Iba*: (A) only wait-for requests are stored by the global deadlock detector (e.g., Fig. 10B); (B) the wait-for requests are registered at the global deadlock detector as each wait-for condition is generated; and (C) there is no need to use time-outs since a check is made for a deadlock each time a wait-for request is registered at the global deadlock detector.

(2) COMPARISON OF CLAIM 1 TO *IBA*

In Claim 1, as amended, "**said step of registering occurs in response to said plurality of participants commencing participation in said distributed transaction**..." Thus, participant data is registered in the name service when the participants commence participating in the distributed transaction, so that later such information can be used by another node, such as in detecting a deadlock. For example, when trying to detect deadlocks, upon expiration of a threshold period of time (such as a time-out), a deadlock handler can determine which participants are involved in the distributed transaction by requesting the participant data from the name service, and then constructing a wait-for graph, thereby avoiding the need to use a broadcast query to all nodes to determine which nodes are participating in the distributed transaction and thus may be involved in the deadlock. (Application, page 12, lines 7-24.)

Note that in the approach of Claim 1: (A) participant data is registered with the name service in response to the participants commencing participation in the distributed transaction, (B) registration occurs without regard to whether a participant has made a wait-for request; and (C) the participant information is requested by a node, such as when such a node wishes

to perform deadlock detection after expiration of a threshold period of time (e.g., such as a time out condition being reached).

Thus, the approach of Claim 1 is fundamentally different from that of *Iba* because in Claim 1, participant data is registered in the name service when the participants commence participation in the distributed transaction, whereas in *Iba*, the approach is to register wait-for relations as the wait-for requests are generated. This means that with the approach of Claim 1, participant data is registered that does not include any wait-for requests, although such wait requests for those participants can be generated later. In contrast, in *Iba*, participant data for participants that have not generated a wait request is not registered.

Thus, the Applicant respectfully submits that *Iba*, either alone or in combination with the other cited references, fails to disclose, teach, suggest, or in any way render obvious registering participant data where "**said step of registering occurs in response to said plurality of participants commencing participation in said distributed transaction...**," as featured in Claim 1.

(3) THE OFFICE ACTION'S CITATIONS IN *IBA*

As noted before, the amendment to Claim 1 adds the feature of "**said step of registering occurs in response to said plurality of participants commencing participation in said distributed transaction...**," which is similar to that of now cancelled Claim 21. With respect to Claim 21, the Office Action states that *Iba* discloses "registering for each participant in said plurality of participants, data that identifies said each participant ***in response to said each participant commencing participation in said distributed transaction*** (Iba: col 2/lines 1-12, col 5/lines 43-50)." (Emphasis added.)

However, the first cited portion from Column 2 of *Iba* merely discusses the general system shown in Fig. 1, which "illustrates a deadlock detection in a closed system." (Col. 2, line 1.) Specifically, this portion of *Iba* states:

> According to the system shown in FIG. 1, deadlocks are detectable between a plurality of transactions generated inside the resource manager 1. That is, when in a system having one resource manage provide with a plurality of access managers 6, and all transactions managed by the transaction manager 4 collectively performs exclusive control to occupy hardware resources when accessed. For this reason, the deadlock detector detects without difficulties

> wait-for relations as to the occupation of hardware resources between a plurality of transactions to be executed inside the resource manager 1. (Col. 2, lines 1-12.)

Thus, this portion of *Iba* is merely describing that deadlocks can easily be detected by deadlock detector 7 of Fig. 1 because deadlock detector 7 is part of resource manager 1 which controls access to all the available resources, such as DB 1, DB 2, and File 1. This portion of the background section of *Iba* is merely discussing what is later referred to in the remainder of *Iba* as local deadlock detection, which *Iba* contrasts with the new global deadlock detection approach that is the subject of *Iba*. As a result, this portion of *Iba* only serves to help illustrate why a local deadlock detector cannot resolve deadlocks arising from global transactions that access resources outside of the reach of a local resource manager, such as resource manager 1 in Fig. 1. There is absolutely nothing in this first portion of *Iba* that relates to registering data for participants **in response to said each participant commencing participation in said distributed transaction**, as featured in previous Claim 21 and the current Claim 1.

Furthermore, the second cited portion from Column 2 of *Iba* cited in the Office Action with respect to Claim 21 states:

> The global deadlock detector is provided with a wait-for graph. The wait-for graph stores wait-for relation between transactions and is used for detecting a deadlock. Contents of the wait-for graph include an address of a node connected to a self node on a shared memory, each node corresponding to each transaction, and a transaction identifier of the self node. The connection relations of each node correspond to mutual wait-for relation between transaction. (Col. 5, lines 43-50.)

Thus, this portion of *Iba* merely states that the GDD uses a WFG that has wait-for relations that include node addresses where the nodes correspond to transactions, with "connection relations" that correspond to the mutual wait-for relations. There is absolutely nothing in this paragraph from *Iba* that has anything to do with registering data for participants **in response to said each participant commencing participation in said distributed transaction**, as featured in previous Claim 21 and the current Claim 1. In fact, this cited portion of *Iba* corresponds to the overall approach discussed above, namely that wait-for relations are tracked by the GDD for detecting deadlocks, but this particular portion of *Iba* in Column 5 is even less specific than the information discussed with respect to Fig. 10A, Fig. 10B, and Fig. 11, all of which have been addressed above.

Therefore, the Applicant is completely at a loss in trying to ascertain what the Office Action could possibly be relying on in either of these two cited portions of *Iba* that allegedly disclose anything about registering data for participants **in response to said each participant commencing participation in said distributed transaction**, as featured in previous Claim 21 and the current Claim 1.

C. CLAIM 1 & *KANAI*

Kanai discloses a "flexible distributed processing system capable of dealing with sophisticated conditions for selecting a server process." (Abstract.) Generally speaking, *Kanai* is directed to an approach for selecting an appropriate server that is providing a desired service in response to a client inquiry. (Abstract) More specifically, *Kanai* teaches "a method of distributed processing among processors, where each processor having a server process for providing services and a service manager for managing the services provided by the server process..." in which the service manager registers each of the services provided by all the server processes along with an executability condition that is used by the service manager to select a server process that provides a service desired by a client such that the selected server process can satisfy the client's request. (Col. 5, lines 12-29.)

However, there is nothing in *Kanai* that discloses, teaches, suggests, or renders obvious registering participant data where "**said step of registering occurs in response to said plurality of participants commencing participation in said distributed transaction...**," as featured in Claim 1. Furthermore, *Kanai* is only concerned with managing different servers and selecting an appropriate server each time a client desires a service from one of the servers, and therefore, *Kanai* is not concerned with a "**distributed transaction**" as featured in Claim 1.

Thus, the Applicant respectfully submits that *Kanai*, either alone or in combination with the other cited references, fails to disclose, teach, suggest, or in any way render obvious registering participant data where "**said step of registering occurs in response to said plurality of participants commencing participation in said *distributed transaction*...**," as featured in Claim 1.

D. CLAIM 1 & *VAN DEN BERG*

Van Den Berg discloses a "distributed processing system that includes a distributed resource manager which detects dependencies between transaction caused by conflicting lock requests." (Abstract.) Generally speaking, *Van Den Berg* is directed to an approach for using wait-for graphs to detect deadlocks in which failure resilience "is achieved by duplicating between agents and servers, rather than by duplicating the servers. As a result, the number of messages between agents and servers in normal operation is not increased." (Abstract.) More specifically, *Van Den Berg* teaches a deadlock detection approach that is similar to *Iba*, namely that dependencies between transaction are "maintained in a queue of lock requests that cannot be immediately granted, and for detecting dependencies between transactions caused by conflicting lock requests (e.g., using a wait-for graph to track wait-for relations and thereby detect deadlocks). (Fig. 3; Col. 3, lines 1-15; Col. 18, lines 4-7.) Again as in *Iba*, the approach in *Van Den Berg* is to only track the lock requests that cannot be granted, not to register participant data in response to the participants commencing participation in the distributed transaction, as featured in Claim 1. As a result, in the approach of *Van Den Berg*, the information being tracked does not include participants that are not waiting on a resource, as is the case in the approach of Claim 1.

Thus, the Applicant respectfully submits that *Van Den Berg*, either alone or in combination with the other cited references, fails to disclose, teach, suggest, or in any way render obvious registering participant data where "**said step of registering occurs in response to said plurality of participants commencing participation in said distributed transaction...**," as featured in Claim 1.

E. CLAIM 1 & *NECHES*

Neches discloses a "system using a sorting network to intercouple multiple processors so as to distribute priority messages to all processors..." (Abstract.) Generally speaking, *Neches* is directed to an approach for achieving greater flexibility as to intercommunications and control by using transaction numbers to identify tasks and track the status of the tasks using prioritized responses and controlling the flow of messages. (Abstract.) More specifically, *Neches* teaches an approach in which the "many message routing, mode control

and status indication functions required for a complex and versatile multiprocessor system are provided in accordance with the invention by a unique combination of message organization and traffic controlling interface circuits functioning with the active logic network." (Col. 3, lines 20-25.) *Neches* uses "transaction identities" and "a single query to all processors" to obtain "the global status of the system." (Col. 3, lines 41-54.) Thus, *Neches* is similar to *Kanai* in that both are generally directed to managing the distribution of tasks among multiple processors and tracking the status of such tasks.

However, there is nothing in *Neches* that discloses, teaches, suggests, or renders obvious registering participant data where "**said step of registering occurs in response to said plurality of participants commencing participation in said distributed transaction...**," as featured in Claim 1. Furthermore, *Neches* is only concerned with message routing, mode control, and status indication functions for a multiprocessor system, and therefore, *Neches* is not concerned with a "**distributed transaction**" as featured in Claim 1.

Thus, the Applicant respectfully submits that *Neches*, either alone or in combination with the other cited references, fails to disclose, teach, suggest, or in any way render obvious registering participant data where "**said step of registering occurs in response to said plurality of participants commencing participation in said *<u>distributed transaction</u>*...**," as featured in Claim 1.

F. CLAIM 1 & *HERRIOT*

Several of the previous Office Actions have relied on U.S. Patent Number 5,862,331 issued to Herriot (" *Herriot* "). While the present Office Action does not rely on *Herriot*, the Applicant wishes to briefly comment on *Herriot's* lack of applicability to Claim 1 as amended above so as to expeditiously advance prosecution of the present Application.

Herriot discloses a "name service system and method for automatic updating on interconnected hosts." (Title.) Generally speaking, *Herriot* is directed to an approach for allowing a name service to operate as a dynamically updated system even though the name service program on the master host is a static system that must be updated manually." (Abstract.) More specifically, *Herriot* teaches the use of a special program called a "proto-server" that carries out dynamic association of program numbers with multiple servers

in a network environment having a static name service. (Col. 3, lines 25-29.) Thus, "the proto-server allows multiple similar servers, or instances of a server, to run under a static name service, without requiring a dedicated pool of program numbers, or a complete upgrade of the service."

However, there is nothing in *Herriot* that discloses, teaches, suggests, or renders obvious registering participant data where "**said step of registering occurs in response to said plurality of participants commencing participation in said distributed transaction...**," as featured in Claim 1. Furthermore, *Herriot* is only concerned with dynamic association of program numbers for servers in a multiple server network , and therefore, *Herriot* is not concerned with a "**distributed transaction**" as featured in Claim 1.

Thus, the Applicant respectfully submits that *Herriot*, either alone or in combination with the other cited references, fails to disclose, teach, suggest, or in any way render obvious registering participant data where "**said step of registering occurs in response to said plurality of participants commencing participation in said <u>*distributed transaction*</u>...**," as featured in Claim 1.

G. THE OFFICE ACTION LACKS THE REQUISITE SUGGESTION, TEACHING, OR MOTIVATION TO COMBINE PRIOR ART REFERENCES

The Office Action states that it would have been obvious to combine *Iba* and *Kanai* in the first rejection of Claim 1 and to combine *Van Den Berg* and *Neches* in the second rejection of Claim 1. However, notwithstanding the fact that none of *Iba*, *Kanai*, *Van Den Berg* or *Neches* disclose registering participant data where "**said step of registering occurs in response to said plurality of participants commencing participation in said <u>*distributed transaction*</u>...**," as featured in Claim 1, the Applicant respectfully submits that there is nothing in any of *Iba*, *Kanai*, *Van Den Berg* or *Neches* that teaches, suggests, or motivates combining their respective teachings.

As stated in the Federal Circuit decision *In re Dembiczak*, 50 USPQ.2d 1617 (Fed. Cir. 1999), (citing *Gore v. Garlock*, 220 USPQ 303, 313 (Fed. Cir. 1983)), "it is very easy to fall victim to the insidious effect of the hindsight syndrome where that which only the inventor taught is used against its teacher." *Id.* The Federal Circuit stated in *Dembiczak* "that

the best defense against subtle but powerful attraction of a hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or suggestion to combine prior art references." *Id.* Thus, the Federal Circuit explains that a proper obviousness analysis requires "***particular factual findings*** regarding the locus of the suggestion, teaching, or motivation to combine prior art references." *Id.* (emphasis added).

In particular, the Federal Circuit states:

> "We have noted that evidence of a suggestion, teaching, or motivation to combine may flow from the prior art references themselves, the knowledge of one of ordinary skill in the art, or, in some cases, from the nature of the problem to be solved...although 'the suggestion more often comes from the teachings of the pertinent references'...The range of sources available, however, does ***not diminish the requirement for actual evidence.*** That is, the ***showing must be clear and particular***...Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence.'" *Id.* (emphasis added; internal citations omitted).

None of *Iba*, *Kanai*, *Van Den Berg* or *Neches* show any suggestion, teaching, or motivation to combine their teachings, nor does the Office Action provide a "clear and particular" showing of the suggestion, teaching, or motivation to combine their teachings. In fact, the only motivation provided in the Office Action are the broad conclusory statements that by combining features of those references, one may achieve the benefits achieved from the invention as described and claimed in the application. It is respectfully submitted that such a hindsight observation is not consistent with the Federal Circuit's requirement for "particular factual findings," and thus that both the rejection of Claim 1 over *Iba* in view of *Kanai* and the rejection of Claim 1 over *Van Den Berg* in view of *Neches* are improper.

H. SUMMARY OF DISCUSSION OF CLAIM 1

Because *Iba*, *Kanai*, *Van Den Berg, Neches*, and *Herriot* fail to disclose, teach, suggest, or in any way render obvious, either alone or in combination, registering participant data where "**said step of registering occurs in response to said plurality of participants commencing participation in said *<u>distributed transaction</u>*...**," as featured in Claim 1, the Applicant respectfully submits that, for at least the reasons stated above, Claim 1 is allowable over the art of record and is in condition for allowance.

I. CLAIMS 11, 22, AND 27

Claims 11, 22, and 26 contain features that are similar to those described above with respect to Claim 1, and in particular all three feature the registering of data "in response to said plurality of participants commencing participation in said distributed transaction" Therefore, based on at least the reasons stated above with respect to Claim 1, the Applicant respectfully submits that Claims 11, 22, and 26 are allowable over the art of record and are in condition for allowance.

J. CLAIMS 2-9, 12-19, 23-26, AND 28-31

Claims 2-9, 12-19, 23-26, and 28-31 are dependent upon Claims 1, 11, 22, and 27, respectively, and thus include each and every feature of the corresponding independent claims. Therefore, it is respectfully submitted that Claims 2-9, 12-19, 23-26, and 28-31 are allowable for the reasons given above with respect to Claims 1, 11, 22, and 27.

CONCLUSION

The Applicant believes that all issues raised in the Office Action have been addressed and that allowance of the pending claims is appropriate. Entry of the amendments and further examination on the merits are respectfully requested.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

For the reasons set forth above, it is respectfully submitted that all of the pending claims are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

To the extent necessary to make this reply timely filed, the Applicant petitions for an extension of time under 37 C.F.R. § 1.136.

If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Craig G. Holmes
Reg. No. 44,770

Date: February 9, 2004

1600 Willow Street
San Jose, CA 95125
Telephone: (408) 414-1080, ext. 207
Facsimile: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

on February 9, 2004 by